

Recherche Opérationnelle

Épisode IV

Coloration et flots

Hervé Hocquard

Université de Bordeaux, France

March 21, 2023

1. Coloration de graphes

Coloration propre des sommets

G : Graphe non-orienté.

Coloration propre

Coloration (propre) = Affectation d'une couleur à chaque sommet de G , de sorte que les sommets adjacents aient des couleurs différentes.

Couleur = État, ressource, etc.

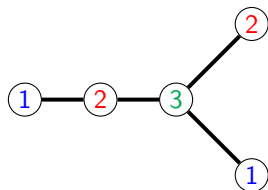
Coloration propre des sommets

G : Graphe non-orienté.

Coloration propre

Coloration (propre) = Affectation d'une couleur à chaque sommet de G , de sorte que les sommets adjacents aient des couleurs différentes.

Couleur = État, ressource, etc.



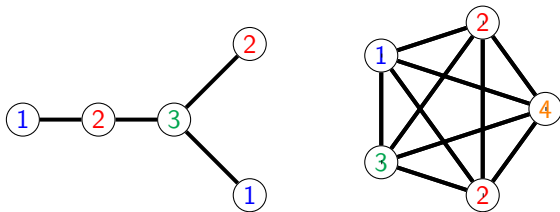
Coloration propre des sommets

G : Graphe non-orienté.

Coloration propre

Coloration (propre) = Affectation d'une couleur à chaque sommet de G , de sorte que les sommets adjacents aient des couleurs différentes.

Couleur = État, ressource, etc.



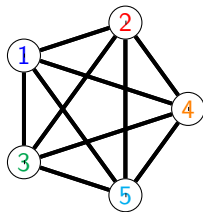
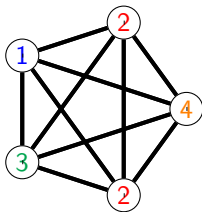
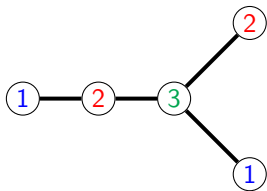
Coloration propre des sommets

G : Graphe non-orienté.

Coloration propre

Coloration (propre) = Affectation d'une couleur à chaque sommet de G , de sorte que les sommets adjacents aient des couleurs différentes.

Couleur = État, ressource, etc.



Remarque : Graphe d'ordre $n \Rightarrow$ Coloration triviale avec n couleurs.
Rarement intéressant d'un point de vue pratique.

Question

Pour G donné, nombre de couleurs **minimum** ?

Remarque : Graphe d'ordre $n \Rightarrow$ Coloration triviale avec n couleurs.
Rarement intéressant d'un point de vue pratique.

Question

Pour G donné, nombre de couleurs **minimum** ?

Nombre chromatique

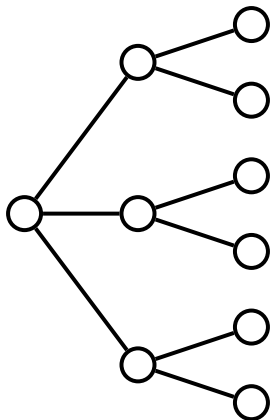
Nombre chromatique $\chi(G)$ de $G =$ Ce minimum.

La remarque implique que $\chi(G) \leq n$ pour tout graphe G d'ordre n .

Notons que $\chi(G) = n$ si G est un graphe complet d'ordre n .

Exemple 1

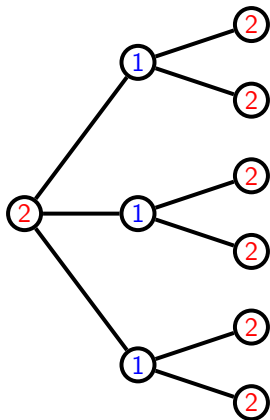
Soit G le graphe suivant :



$$\chi(G) = ?$$

Exemple 1

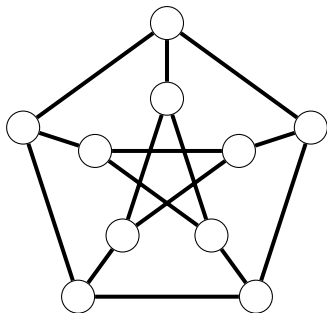
Soit G le graphe suivant :



$$\chi(G) = 2$$

Exemple 2

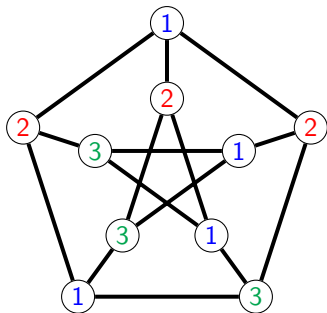
Soit G le graphe suivant :



$$\chi(G) = ?$$

Exemple 2

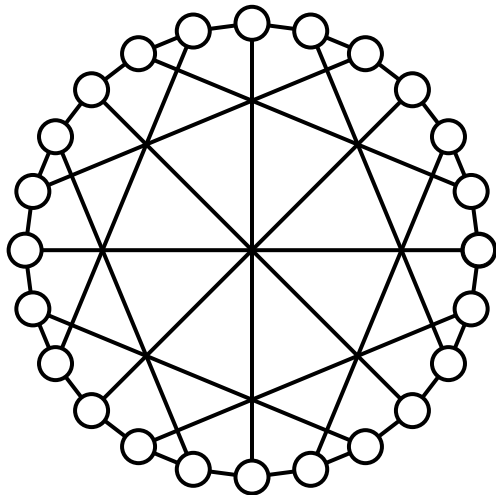
Soit G le graphe suivant :



$$\chi(G) = 3$$

Exemple 3

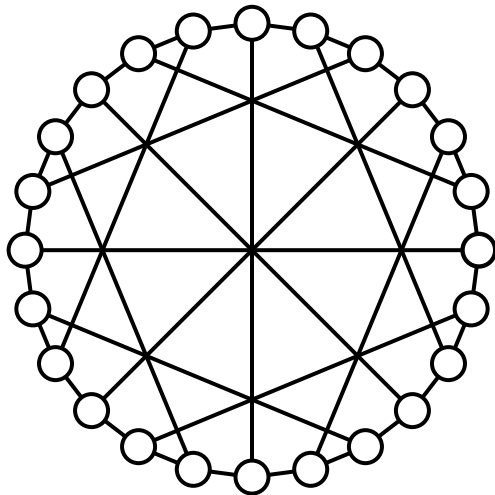
Soit G le graphe suivant :



$$\chi(G) = ???????$$

Exemple 3

Soit G le graphe suivant :



$\chi(G) = 3$ en fait...

Bornes sur $\chi(G)$

Astuces pour se faire une idée de $\chi(G)$:

Proposition (folklore)

Pour tout graphe G d'ordre n , on a

$$\omega(G) \leq \chi(G) \leq \Delta(G) + 1,$$

où ω est l'ordre d'un plus grand sous-graphe complet de G , et $\Delta(G)$ est le plus grand degré d'un sommet de G .

Bornes sur $\chi(G)$

Astuces pour se faire une idée de $\chi(G)$:

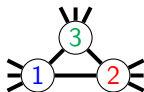
Proposition (folklore)

Pour tout graphe G d'ordre n , on a

$$\omega(G) \leq \chi(G) \leq \Delta(G) + 1,$$

où ω est l'ordre d'un plus grand sous-graphe complet de G , et $\Delta(G)$ est le plus grand degré d'un sommet de G .

- $\omega(G) \leq \chi(G)$:



Bornes sur $\chi(G)$

Astuces pour se faire une idée de $\chi(G)$:

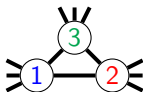
Proposition (folklore)

Pour tout graphe G d'ordre n , on a

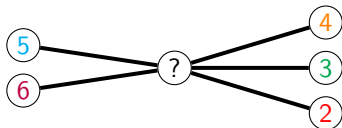
$$\omega(G) \leq \chi(G) \leq \Delta(G) + 1,$$

où ω est l'ordre d'un plus grand sous-graphe complet de G , et $\Delta(G)$ est le plus grand degré d'un sommet de G .

- $\omega(G) \leq \chi(G)$:



- $\chi(G) \leq \Delta(G) + 1$:



Bornes sur $\chi(G)$

Astuces pour se faire une idée de $\chi(G)$:

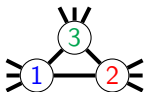
Proposition (folklore)

Pour tout graphe G d'ordre n , on a

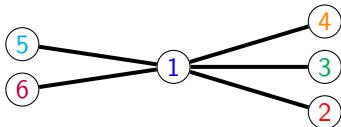
$$\omega(G) \leq \chi(G) \leq \Delta(G) + 1,$$

où ω est l'ordre d'un plus grand sous-graphe complet de G , et $\Delta(G)$ est le plus grand degré d'un sommet de G .

- $\omega(G) \leq \chi(G)$:

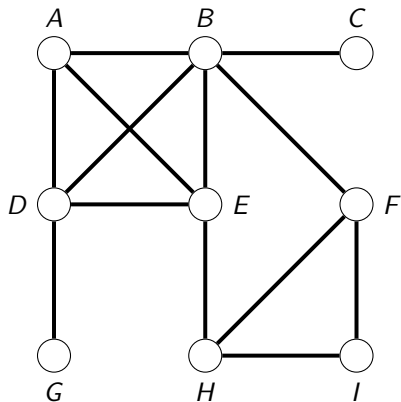


- $\chi(G) \leq \Delta(G) + 1$:



Exemple

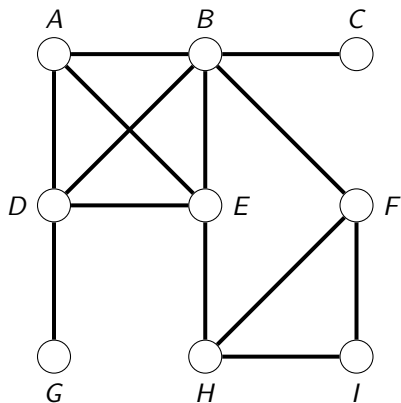
Soit G le graphe suivant :



- $\omega(G) = ?$
- $\Delta(G) = ?$

Exemple

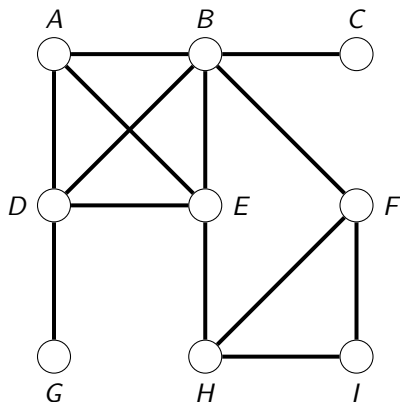
Soit G le graphe suivant :



- $\omega(G) = 4$ (A, B, D, E)
- $\Delta(G) = 5$ (B)

Exemple

Soit G le graphe suivant :



- $\omega(G) = 4$ (A, B, D, E)
- $\Delta(G) = 5$ (B)

$$\Rightarrow \chi(G) \in \{4, 5, 6\}$$

Algorithme de coloration de Welsh-Powell

Idée : Coloration “gloutonne”. On commence par les sommets de grands degrés, car ils sont les plus “dangereux”.

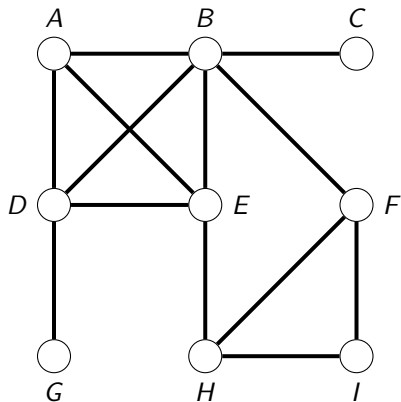
Idee : Coloration “gloutonne”. On commence par les sommets de grands degrés, car ils sont les plus “dangereux”.

Algorithme de Welsh-Powell :

- **Entrée :** Un graphe G .
- **Initialisation :** Ordonner les sommets par l'ordre décroissant de leurs degrés.
- **Traitement :**
 - Considérer la couleur $\alpha = 1$. Puis :
 - 1 Parcourir les sommets suivant l'ordre fixé.
 - 2 Pour chaque sommet considéré, on lui attribue la couleur α si possible.
 - Si tous les sommets ne sont pas coloriés, on répète avec la couleur α suivante.
- **Sortie :** Une coloration de G .

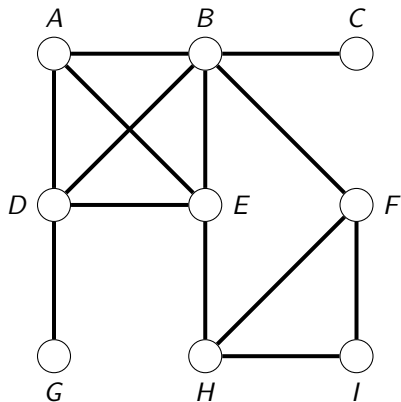
Exemple 1

Soit G le graphe suivant :



Exemple 1

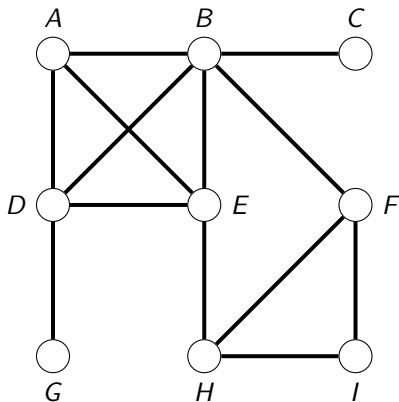
Soit G le graphe suivant :



Étape 1 : On ordonne les sommets suivant leurs degrés.

Exemple 1

Soit G le graphe suivant :



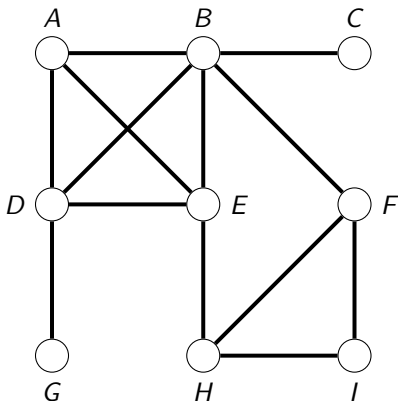
Étape 1 : On ordonne les sommets suivant leurs degrés.

Sommet	B	D	E	A	F	H	I	C	G
Degré	5	4	4	3	3	3	2	1	1

Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur 1 dès que possible.

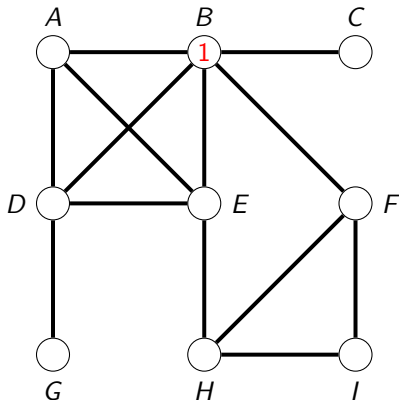
Ordre : $B - D - E - A - F - H - I - C - G$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

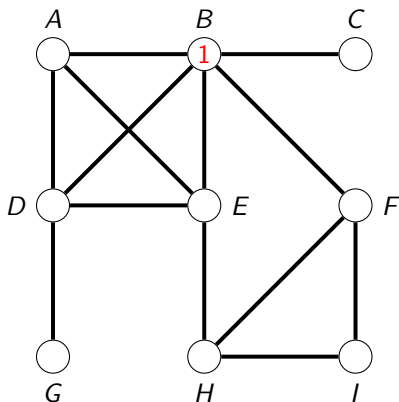
Ordre : B - D - E - A - F - H - I - C - G



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

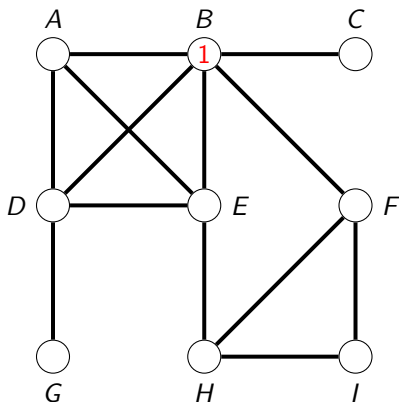
Ordre : $B - \underline{D} - E - A - F - H - I - C - G$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

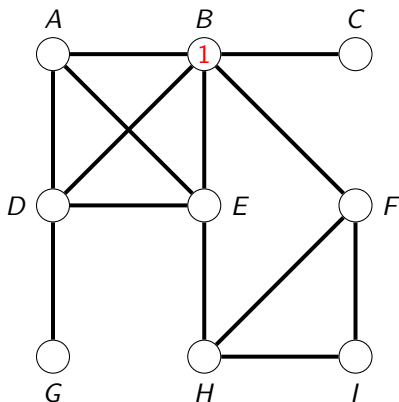
Ordre : $B - D - \underline{E} - A - F - H - I - C - G$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

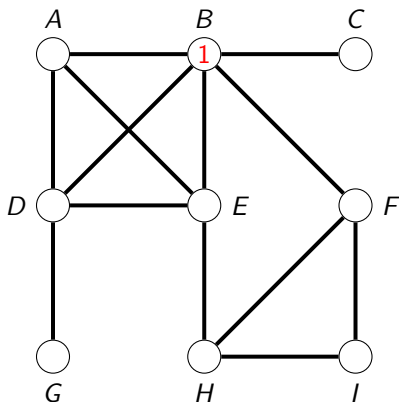
Ordre : $B - D - E - \underline{A} - F - H - I - C - G$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

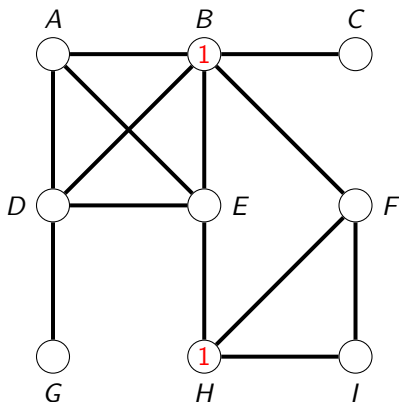
Ordre : $B - D - E - A - \underline{F} - H - I - C - G$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

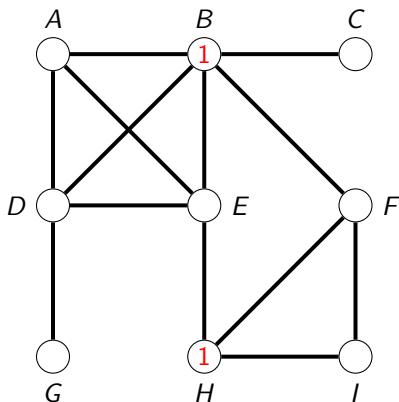
Ordre : $B - D - E - A - F - \underline{H} - I - C - G$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

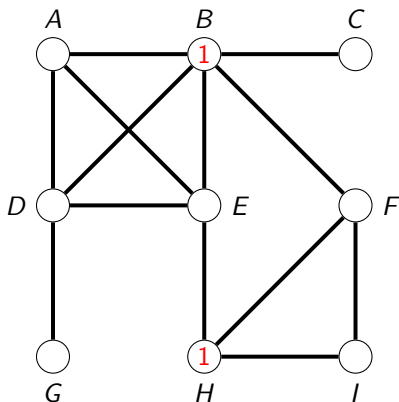
Ordre : $B - D - E - A - F - H - \underline{I} - C - G$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

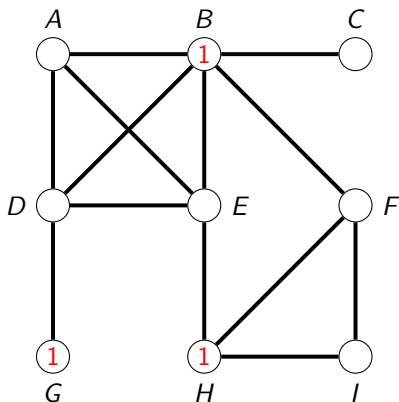
Ordre : $B - D - E - A - F - H - I - \underline{C} - G$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

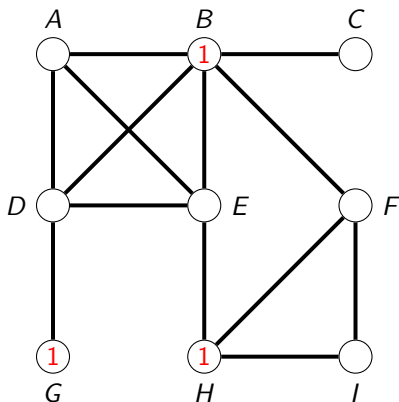
Ordre : $B - D - E - A - F - H - I - C - \underline{G}$



Exemple 1

Étape 2 (1) : En parcourant les sommets dans l'ordre, on attribue la couleur **1** dès que possible.

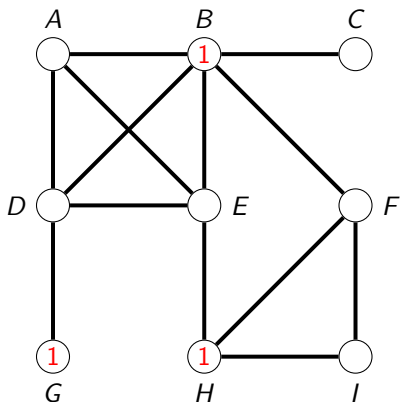
Ordre : $B - D - E - A - F - H - I - C - G$



Exemple 1

Étape 2 (2) : Pareil, mais avec la couleur 2.

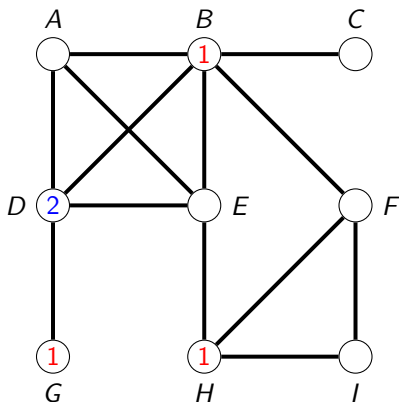
Ordre : $B - D - E - A - F - H - I - C - G$



Exemple 1

Étape 2 (2) : Pareil, mais avec la couleur 2.

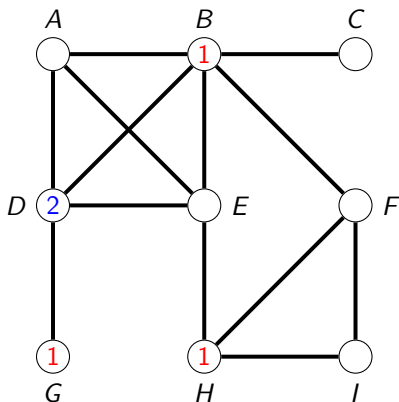
Ordre : $B - \underline{D} - E - A - F - H - I - C - G$



Exemple 1

Étape 2 (2) : Pareil, mais avec la couleur 2.

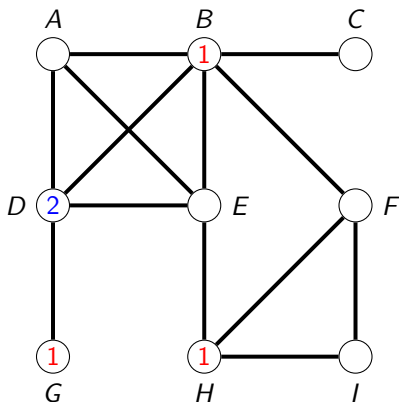
Ordre : $B - D - \underline{E} - A - F - H - I - C - G$



Exemple 1

Étape 2 (2) : Pareil, mais avec la couleur 2.

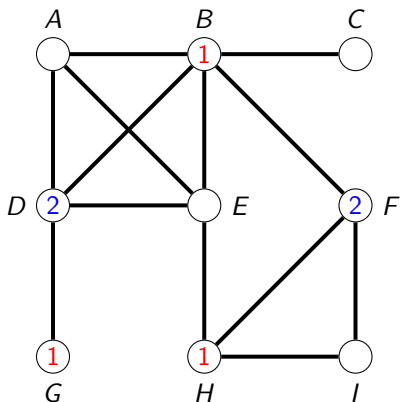
Ordre : $B - D - E - \underline{A} - F - H - I - C - G$



Exemple 1

Étape 2 (2) : Pareil, mais avec la couleur 2.

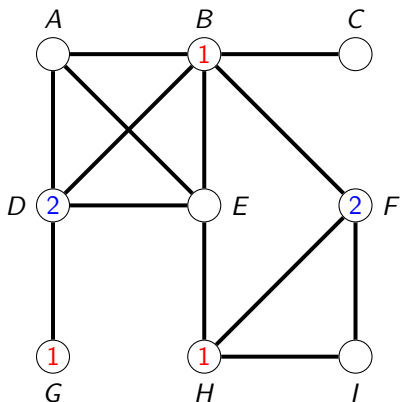
Ordre : $B - D - E - A - \underline{E} - H - I - C - G$



Exemple 1

Étape 2 (2) : Pareil, mais avec la couleur 2.

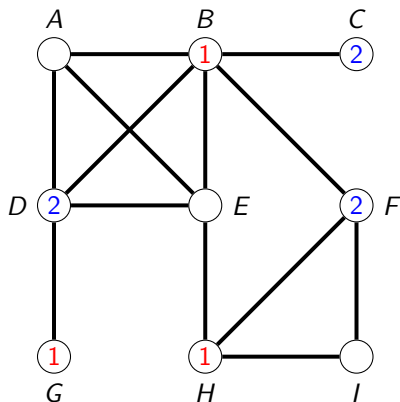
Ordre : $B - D - E - A - F - H - \underline{I} - C - G$



Exemple 1

Étape 2 (2) : Pareil, mais avec la couleur 2.

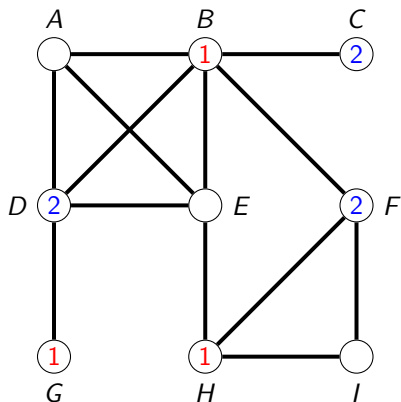
Ordre : $B - D - E - A - F - H - I - \underline{C} - G$



Exemple 1

Étape 2 (2) : Pareil, mais avec la couleur 2.

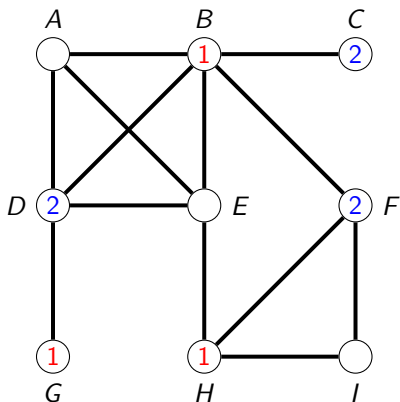
Ordre : $B - D - E - A - F - H - I - C - G$



Exemple 1

Étape 2 (3) : La couleur 3 maintenant...

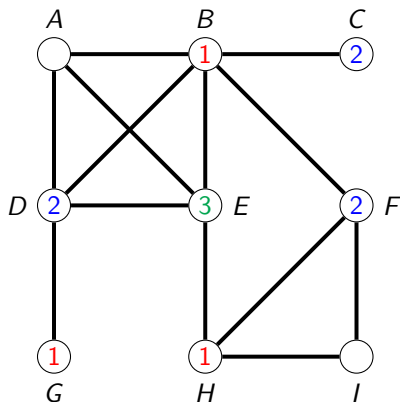
Ordre : *B - D - E - A - F - H - I - C - G*



Exemple 1

Étape 2 (3) : La couleur 3 maintenant...

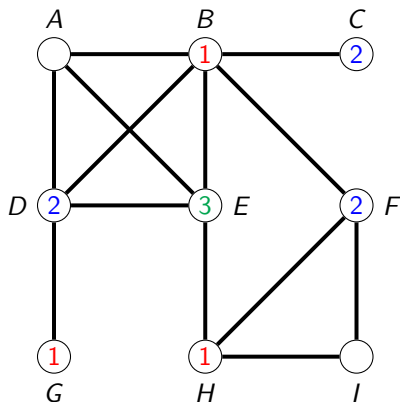
Ordre : $B - D - \underline{E} - A - F - H - I - C - G$



Exemple 1

Étape 2 (3) : La couleur 3 maintenant...

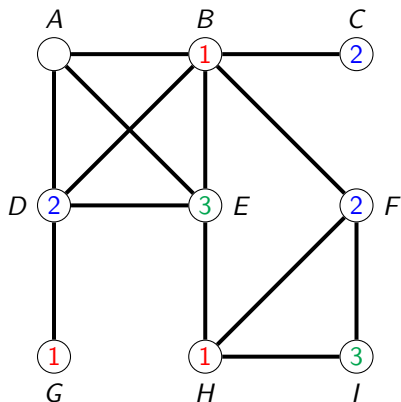
Ordre : $B - D - E - \underline{A} - F - H - I - C - G$



Exemple 1

Étape 2 (3) : La couleur 3 maintenant...

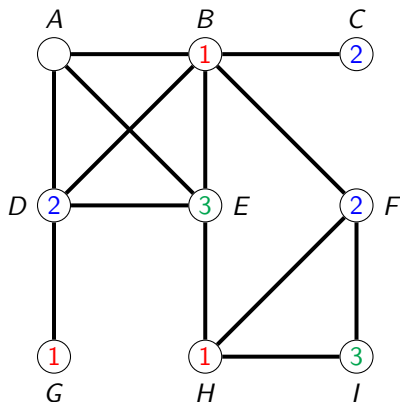
Ordre : $B - D - E - A - F - H - \underline{I} - C - G$



Exemple 1

Étape 2 (3) : La couleur 3 maintenant...

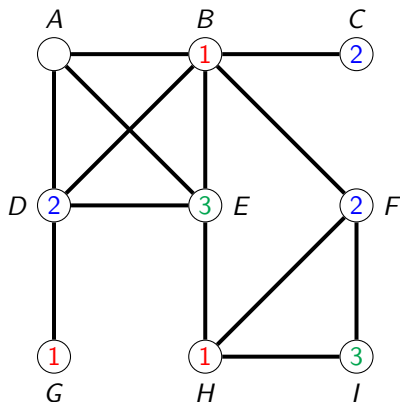
Ordre : *B - D - E - A - F - H - I - C - G*



Exemple 1

Étape 2 (4) : Puis la couleur 4.

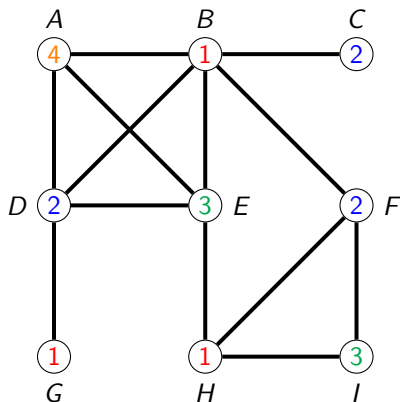
Ordre : $B - D - E - A - F - H - I - C - G$



Exemple 1

Étape 2 (4) : Puis la couleur 4.

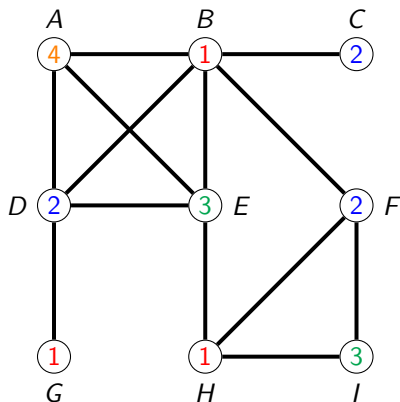
Ordre : $B - D - E - \underline{A} - F - H - I - C - G$



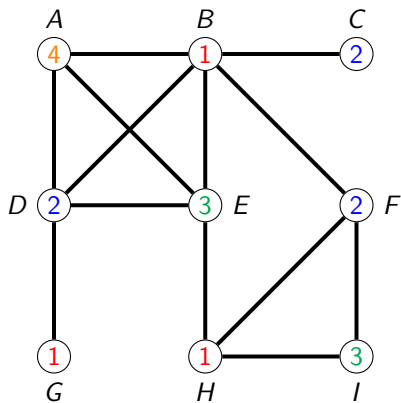
Exemple 1

Étape 2 (4) : Puis la couleur 4.

Ordre : $B - D - E - A - F - H - I - C - G$

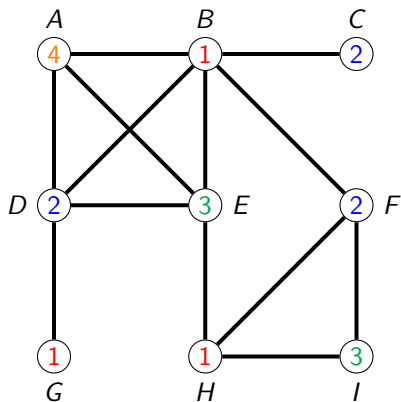


Exemple 1



Donc $\chi(G) \leq 4$. Optimal ?

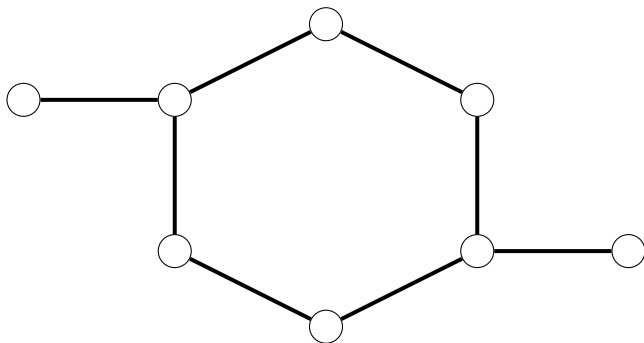
Exemple 1



Donc $\chi(G) \leq 4$. Optimal ? *oui, car $\omega(G) = 4$!*

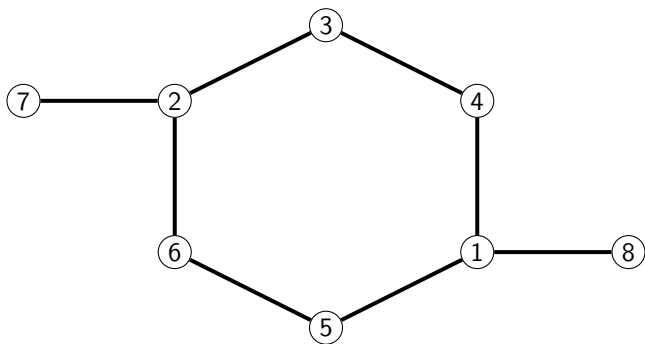
Exemple 2

Et si G est le graphe suivant ?



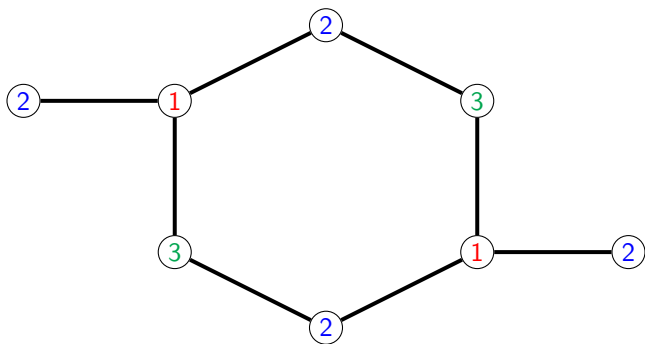
Exemple 2

Et si G est le graphe suivant ?



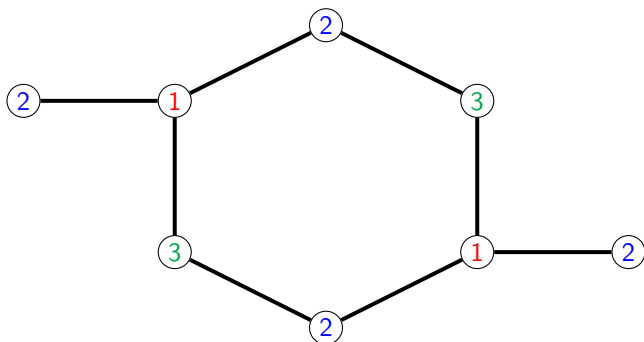
Exemple 2

Et si G est le graphe suivant ?



Exemple 2

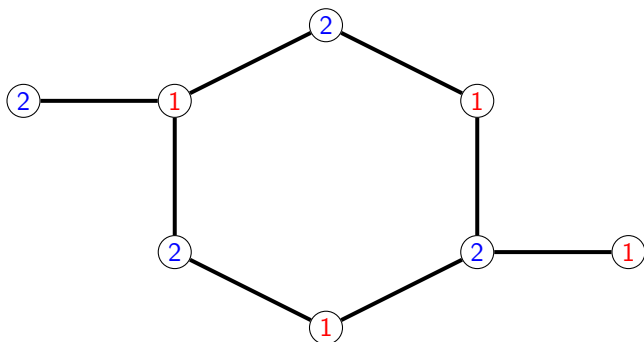
Et si G est le graphe suivant ?



Optimal ?

Exemple 2

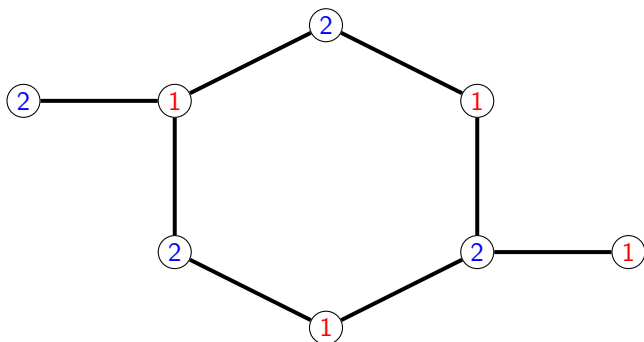
Et si G est le graphe suivant ?



Optimal ? *non, car $\chi(G) = 2$!*

Exemple 2

Et si G est le graphe suivant ?



Optimal ? *non, car $\chi(G) = 2$!*

L'ordre des degrés joue un rôle primordial !

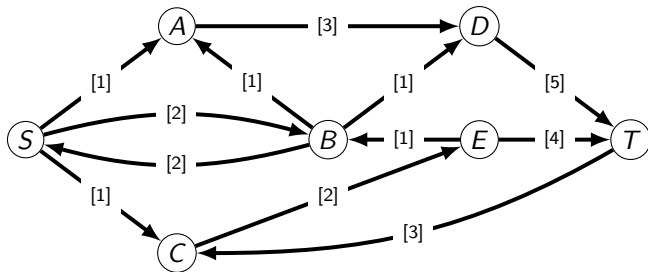
2. Flots

Réseau de transport

Un **réseau** (de transport) est un graphe orienté, pondéré, connexe, sans boucle, où chaque arête est dotée d'une **capacité** $[x]$. Ici, **nœud** = sommet. On a, dans le réseau, deux nœuds particuliers S et T , qui sont la **source** et le **puits** du réseau. Les autres nœuds sont dits **intermédiaires**.

Réseau de transport

Un **réseau** (de transport) est un graphe orienté, pondéré, connexe, sans boucle, où chaque arête est dotée d'une **capacité** $[x]$. Ici, **nœud** = sommet. On a, dans le réseau, deux nœuds particuliers S et T , qui sont la **source** et le **puits** du réseau. Les autres nœuds sont dits **intermédiaires**.



Ici, 5 nœuds intermédiaires. La capacité de l'arc $E \rightarrow T$ est de 4.

Flot

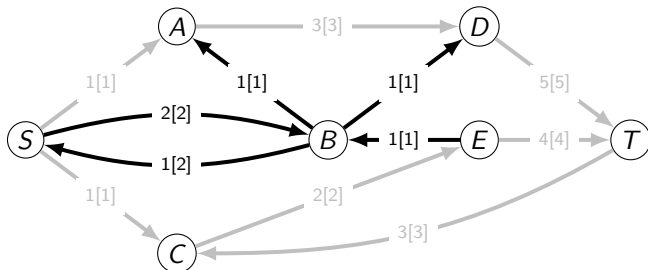
Flot = Acheminement d'une certaine quantité de flux de S vers T via les arcs du réseau, vérifiant :

- 1 la quantité de flux transitant via un arc ne doit pas excéder sa capacité ;
- 2 le flot de tout nœud inter. $V \notin \{S, T\}$ doit être nul, i.e. la quantité $F_e(V)$ y entrant doit être égale à la quantité $F_s(V)$ en sortant ;

Flot

Flot = Acheminement d'une certaine quantité de flux de S vers T via les arcs du réseau, vérifiant :

- 1 la quantité de flux transitant via un arc ne doit pas excéder sa capacité ;
- 2 le flot de tout nœud inter. $V \notin \{S, T\}$ doit être nul, i.e. la quantité $F_e(V)$ y entrant doit être égale à la quantité $F_s(V)$ en sortant ;

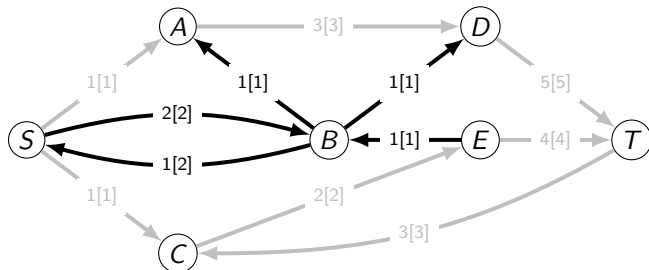


ok pour B : $F_e(B) = F_s(B) = 3$.

Flot

Flot = Acheminement d'une certaine quantité de flux de S vers T via les arcs du réseau, vérifiant :

- 1 la quantité de flux transitant via un arc ne doit pas excéder sa capacité ;
- 2 le flot de tout nœud inter. $V \notin \{S, T\}$ doit être nul, i.e. la quantité $F_e(V)$ y entrant doit être égale à la quantité $F_s(V)$ en sortant ;



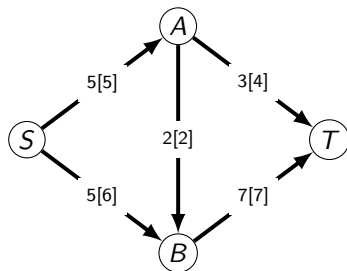
ok pour B : $F_e(B) = F_s(B) = 3$. mais pas flot !

Flot maximum

- Flot **maximum** = Acheminement maximum.
- Flot **complet** = Sur chaque chemin de S vers T , un arc saturé.

Flot maximum

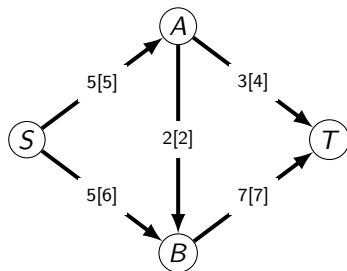
- Flot **maximum** = Acheminement maximum.
- Flot **complet** = Sur chaque chemin de S vers T , un arc saturé.



flot complet (à cause des arcs saturés $S \rightarrow A$ et $B \rightarrow T$), donc maximal.

Flot maximum

- Flot **maximum** = Acheminement maximum.
- Flot **complet** = Sur chaque chemin de S vers T , un arc saturé.



flot complet (à cause des arcs saturés $S \rightarrow A$ et $B \rightarrow T$), donc maximal.

Un flot maximum est complet, mais la réciproque n'est pas vraie.

Algorithme de Ford-Fulkerson

Idée : Faire transiter des gens le long de chemins non-saturés.

Remarque : Cela ne viole pas les contraintes de flot.

Algorithme de Ford-Fulkerson

Idée : Faire transiter des gens le long de chemins non-saturés.

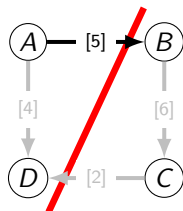
Remarque : Cela ne viole pas les contraintes de flot.

Coupe d'un graphe

On sépare les sommets en deux parts V_1, V_2 .

Coupe = Arcs de même direction d'une part à l'autre.

Capacité de la coupe = Somme des capacités de ces arcs.



coupe de capacité 5

Algorithme de Ford-Fulkerson

Idée : Faire transiter des gens le long de chemins non-saturés.

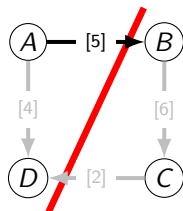
Remarque : Cela ne viole pas les contraintes de flot.

Coupe d'un graphe

On sépare les sommets en deux parts V_1, V_2 .

Coupe = Arcs de même direction d'une part à l'autre.

Capacité de la coupe = Somme des capacités de ces arcs.



coupe de capacité 5

Remarque : Le flot max ne peut pas excéder la capacité d'une coupe.

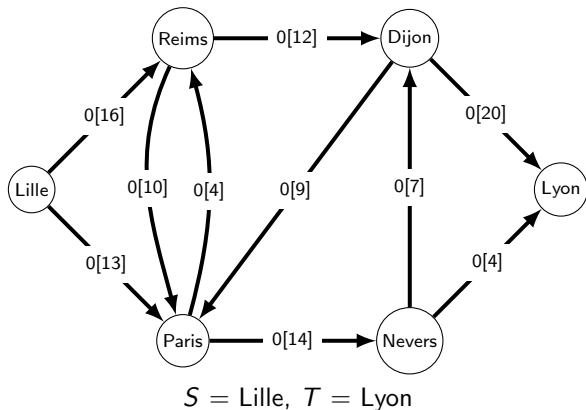
Exemple

Une usine localisée à Lille produit des voitures. Ces voitures sont acheminées en train jusqu'à Lyon où elles sont stockées dans un entrepôt puis vendues. Les capacités de transport des trains sont les suivantes :

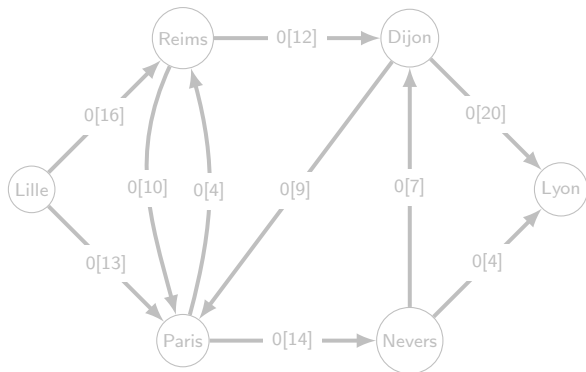
- *Ligne Lille–Reims : 16 voitures par jour.*
- *Ligne Lille–Paris : 13 voitures par jour.*
- *Ligne Paris–Reims : 4 voitures par jour.*
- *Ligne Reims–Paris : 10 voitures par jour.*
- *Ligne Reims–Dijon : 12 voitures par jour.*
- *Ligne Paris–Nevers : 14 voitures par jour.*
- *Ligne Dijon–Paris : 9 voitures par jour.*
- *Ligne Nevers–Dijon : 7 voitures par jour.*
- *Ligne Nevers–Lyon : 4 voitures par jour.*
- *Ligne Dijon–Lyon : 20 voitures par jour.*

Acheminer le plus de voitures possible par jour ??

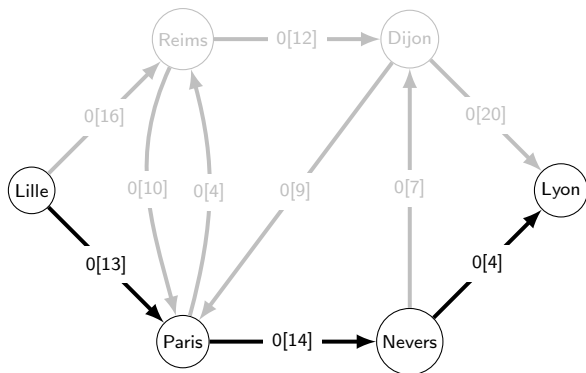
Étape 1 : Modéliser la situation par un réseau.



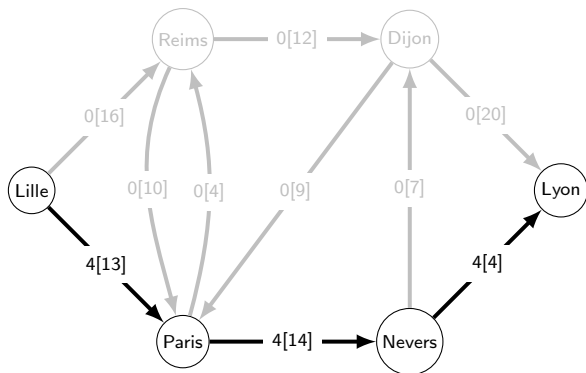
Étape 2 : On choisit une chaîne non-saturée de S à T , et on y fait transiter le maximum de voitures.



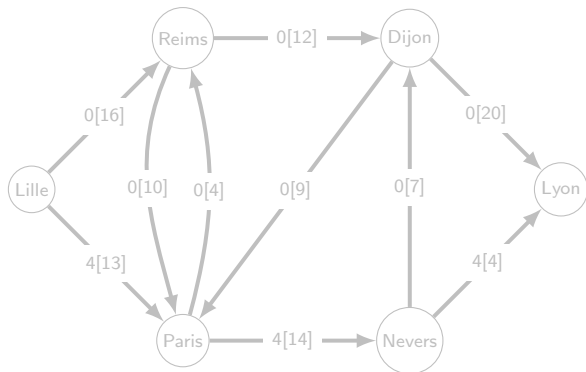
Étape 2 : On choisit une chaîne non-saturée de S à T , et on y fait transiter le maximum de voitures.



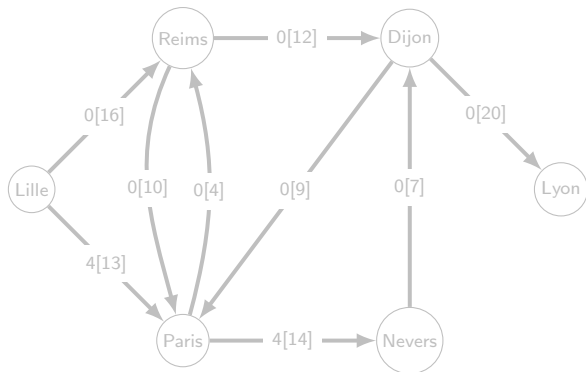
Étape 2 : On choisit une chaîne non-saturée de S à T , et on y fait transiter le maximum de voitures.



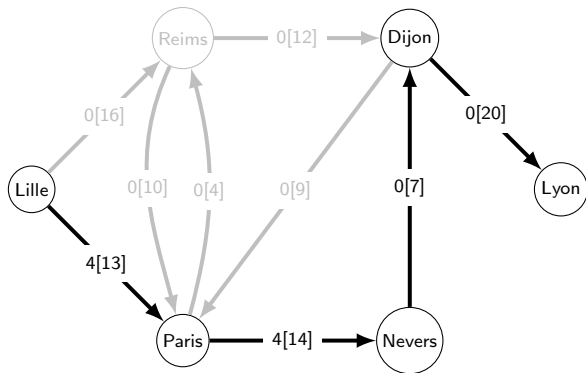
Étape 2 : On choisit une chaîne non-saturée de S à T , et on y fait transiter le maximum de voitures.



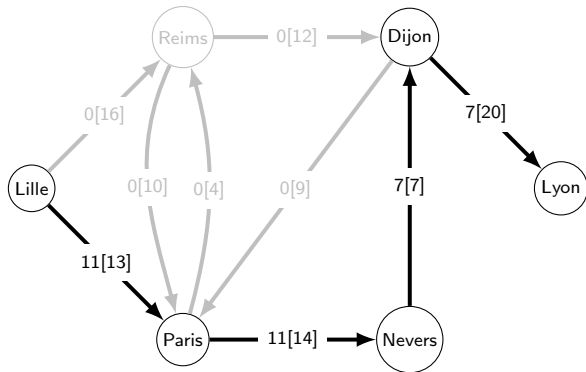
Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.



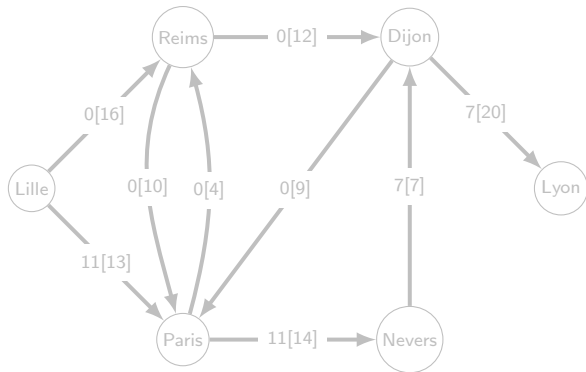
Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.



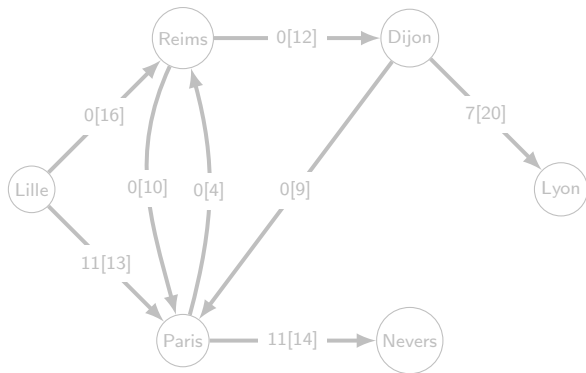
Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.



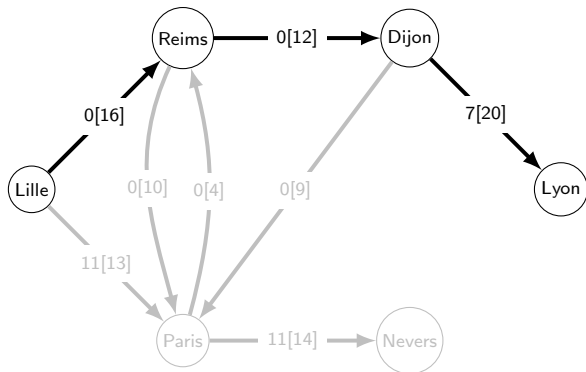
Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.



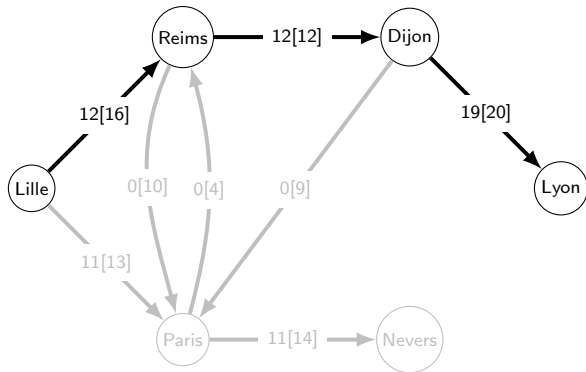
Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.



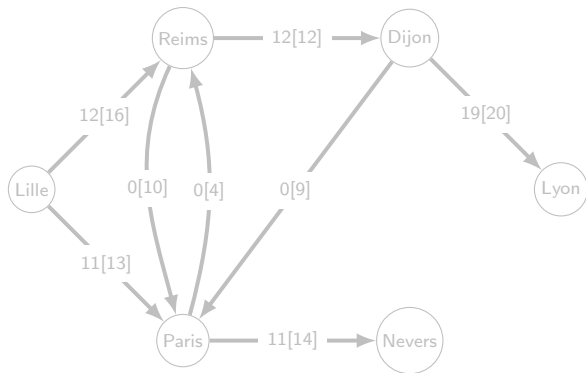
Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.



Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.

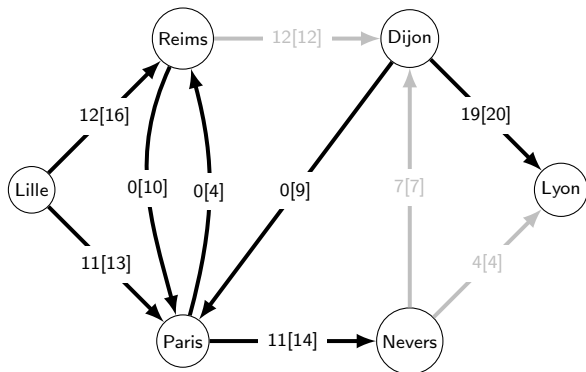


Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.



Solution

Étape 2 : On répète tant qu'on trouve un chemin de S à T non-saturé.

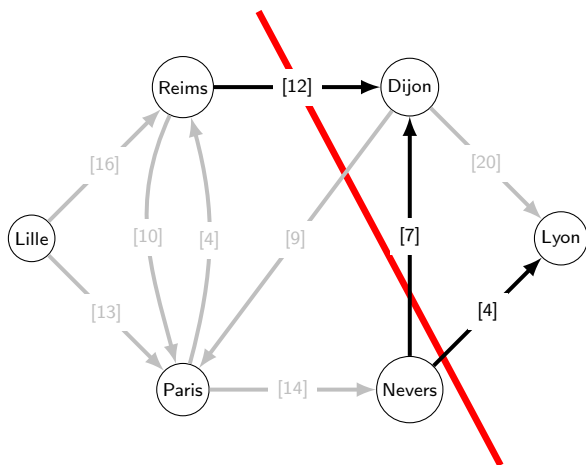


Plus de chemins non-saturés de S à T → fini.

Valeur du flot : 23.

Optimalité ?

Coupe ?



coupe de capacité 23.

Or $\text{flot max} \leq \text{coupe min}$. **Donc pas mieux que 23 !**